Amendment To The Specification:

Please replace the title with the following rewritten title:

System and Method for Using Multiple Working Memories to Improve Microprocessor Security

Please replace the paragraph beginning at page 9, line 16, with the following rewritten paragraph:

--The two other lines (534, 535) each lock one of the two blocks ~~locks~~ or stacks or memorization registers (54, 55). Each of these blocks has a number of memorization registers (54) and respectively (55) for the storage of the information which will be described below. These registers (54, 55) are connected to the bus (3) common to the memories. This bus (3) is used for loading the switching circuit (53) with the values needed to render the control lines (531A, 532A, 532D, 536A, 536D, 533, 534, 535) active or inactive depending on the desired operating mode. The non-volatile memory (6) contains the circuit operating system and an initial application program (P1) called subsequently the main program and a second program (P2) called subsequently the secondary program, the sequencer (8), the registers (R2), the timer (R3) and the random generator (R1) also connected to the bus ~~(P3)~~ (3) and the three elements (R1, R2, R3) connected to an interrupt generator circuit (4) connected to the processor interrupt inputs (1) using ~~on~~ the Interrupt Enable Register (IER) of the processor, one of the bits which is generally reserved and available for applications specific to some users.--

Please replace the paragraph beginning on Page 10, line 11, with the following paragraph:

--In an initial embodiment, main program (P1) contained in the non-volatile memory (6) modifies as necessary the state of switching circuit (53) through bus (3), a process that does not represent any difficulties in execution. Momentarily, this switches out the main working RAM (51) or part of this memory by acting on the CE (Chip Enable) input validating a memory package and all the registers needed for the first block (54) for return to normal operation. These memories and registers can be advantageously of the static type so as to save on the energy needed for maintaining them. The switching circuit (53), therefore, replaces the dummy memory (52) for the main working memory (51) so that the programs are executed using exclusively the dummy memory instead of the main working memory. Said dummy memory (52) can be at the same addresses as the memory for which it is replaced but can also be at a different address. One advantageous and economical solution consists in using a very small RAM for this dummy memory. Indeed, this dummy memory does not play a functional part for the main program, and the addressable space can be restricted by simply decreasing the length of the addressing register (A3). It is also possible to "fold back" the address on itself by setting up an Exclusive OR between several address register blocks. Thus, if the addressable space of the main working memory is 512 bytes, the dummy memory can be restricted to 32 bytes without difficulty, thus leading to a very economical solution. The 32 bytes can correspond, for instance, to the simple addition of a RAM memory line to the matrix of the main working memory. In this case, this line will have its own address registers (A3) and ~~failure~~ data registers (D3). Then the switching circuit (53) activates the dummy memory, it can inhibit any write access to the NVM so as not to disturb its content. --

Please replace the paragraph beginning on Page 13, line 1 with the following

paragraph:

--Figure 2 illustrates operation in the interrupt mode. The diagram shows that the first interrupt pulse IT, transmitted by the interrupt circuit on line (31) toward the processing unit (1), is not taken into consideration because it was masked by means of the register and the interrupt masking using the instructions "MOVE immediate data to register IER" so as to load the data into the masking register. It is assumed that the current instruction unmasks the diversion interruption (but this can be done by any other instruction at a different time). In this case, the second pulse is considered by the processing unit (1) causing the switching circuit (53) to switch over and, accordingly, the second block of registers (55) and the DumRAM (52) become active instead of the first block (54) and the ~~RAM dummy~~ main working memory (51). It is to be noted that the acknowledging of the interruption is only possible during the transition from one state to another, for instance between (S2) and (S3) so as to memorize a stable and consistent state of the machine and, above all, to restore exactly the same state when the interrupted program returns. If this interruption is acknowledged, as is the habitual case at the end of an instruction, there is no particular problem when the interrupted program is recovered because it takes place normally on the next instruction. Conversely, if the interruption occurs during the execution of an instruction, for example in state (S2), it is obviously necessary for the sequencing circuits to be reestablished identically so as to correctly trigger the state (S3) on recovery of the interrupted program. This can be achieved, for instance, by a direct link between the register (T11) and the sequencer (8) via bus (3) at the moment of recovery. This link can also be specific without going via the bus ~~(S3)~~. It might also be advantageous to include the status memorization registers in the sequencer itself to avoid the mobilization of the bus during this phase.--

Replace the paragraph beginning on Page 18, line 6 with the following replacement paragraph:

C4

--In a third embodiment, the random generator (2) can be interrogated by the processor (1) via the bus (3) by a read instruction in order to find out its status, or by directly reading a determined pulse or by grouping several of them or yet again by considering the content of register (R2) loaded from random generator ~~92)~~ (2). When the main program needs protection, it transfers control to the secondary program in a similar way to the mechanism seen previously.--